Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A computer-implemented method for identifying, in a device space, an effective centerscan object color along an edge between an overscan object and a centerscan object, the overscan object having a higher paint order than the centerscan object, the method comprising:

mapping the edge to the device space;

identifying a set of overscan boundary pixels in the device space, the overscan boundary pixels being device space pixels that are intersected by the edge;

identifying a set of centerscan boundary pixels in the device space, each centerscan boundary pixel being a device space pixel on the centerscan object side of the edge that is adjacent to an overscan boundary pixel; and

~~creating a vector pointing in a direction of the centerscan object relative to the edge;~~

~~applying the vector to each overscan boundary pixel in the set of overscan boundary pixels to identify a corresponding set of centerscan boundary pixels in the device space; and~~

mapping each centerscan boundary pixel to the centerscan object to identify a color of the centerscan boundary pixel.

2. (Original) The method of claim 1, wherein the centerscan object is a raster image and the overscan object is a vector object.

3. (Currently Amended) The method of claim 2, wherein ~~an image resolution differs from a device resolution~~ a size of pixels in the centerscan object is different than a size of the centerscan boundary pixels when each centerscan boundary pixel is mapped to the centerscan object.

4. (Currently Amended) The method of claim 1, wherein ~~identifying a~~ mapping each centerscan boundary pixel to the centerscan object to identify the color of the centerscan boundary pixel comprises:

assigning a color to ~~coloring~~ the centerscan boundary pixel ~~in the device space in accordance with a center scan rule~~ using a color of the centerscan object at the center of the centerscan boundary pixel.

5. (Currently Amended) The method of claim 1, wherein ~~creating a vector comprises~~ identifying the set of centerscan boundary pixels includes:

creating a vector pointing in a direction of the centerscan object relative to the edge, the vector being specified in device pixels; and

applying the vector to each overscan boundary pixel in the set of overscan boundary pixels.

6. (Currently Amended) The method of claim 1, wherein ~~creating a vector comprises~~ identifying the set of centerscan boundary pixels includes:

creating a vector pointing in a direction of the centerscan object relative to the edge, the vector being normal to the edge; and

applying the vector to each overscan boundary pixel in the set of overscan boundary pixels.

7. (Currently Amended) The method of claim 1, wherein ~~creating a vector comprises~~ identifying the set of centerscan boundary pixels includes:

creating a vector pointing in a direction of the centerscan object relative to the edge, the vector being normal to an axis in the device space; and

applying the vector to each overscan boundary pixel in the set of overscan boundary pixels.

8. (Cancelled)

9. (Currently Amended) The method of claim 1, further comprising:

identifying one or more subsections, each subsection including one or more contiguous centerscan boundary pixels having the same color, to be used in trapping.

10. (Currently Amended) A computer program product, residing on a computer-readable medium, for identifying, in a device space, an effective centerscan object color along an edge between an overscan object and a centerscan object, the overscan object having a higher paint order than the centerscan object, the computer program product ~~containing~~ comprising instructions for causing a computer to:

map the edge to the device space;

identify a set of overscan boundary pixels in the device space, the overscan boundary pixels being device space pixels that are intersected by the edge;

identify a set of centerscan boundary pixels in the device space, each centerscan boundary pixel being a device space pixel on the centerscan object side of the edge that is adjacent to an overscan boundary pixel; and

~~create a vector pointing in a direction of the centerscan object relative to the edge;~~

~~apply the vector to each overscan boundary pixel in the set of overscan boundary pixels to identify a corresponding set of centerscan boundary pixels in the device space; and~~

map each centerscan boundary pixel to the centerscan object to identify a color of the centerscan boundary pixel.

11. (Original) The computer program product of claim 10, wherein the centerscan object is a raster image and the overscan object is a vector object.

12. (Currently Amended) The computer program product of claim 11, wherein ~~an image resolution differs from a device resolution~~ a size of pixels in the centerscan object is different than a size of the centerscan boundary pixels when each centerscan boundary pixel is mapped to the centerscan object.

13. (Currently Amended) The computer program product of claim 10, wherein the computer program further includes instructions for causing a computer to:

assign a color to each [[the]] centerscan boundary pixel ~~in the device space in accordance with a center scan rule~~ using a color of the centerscan object at the center of the respective centerscan boundary pixel.

14. (Currently Amended) The computer program product of claim 10, wherein the ~~computer program further includes~~ instructions for causing a computer to identify the set of centerscan boundary pixels include instructions for causing a computer to:

create a vector pointing in a direction of the centerscan object relative to the edge, the vector being specified in device pixels; and

apply the vector to each overscan boundary pixel in the set of overscan boundary pixels.

15. (Currently Amended) The computer program product of claim 10, wherein the ~~computer program further includes~~ instructions for causing a computer to identify the set of centerscan boundary pixels include instructions for causing a computer to:

create a vector pointing in a direction of the centerscan object relative to the edge, the vector being normal to the edge; and

apply the vector to each overscan boundary pixel in the set of overscan boundary pixels.

16. (Currently Amended) The computer program product of claim 10, wherein the ~~computer program further includes~~ instructions for causing a computer to identify the set of centerscan boundary pixels include instructions for causing a computer to:

create a vector pointing in a direction of the centerscan object relative to the edge, the vector being normal to an axis in the device space; and

apply the vector to each overscan boundary pixel in the set of overscan boundary pixels.

17. (Cancelled)

18. (Currently Amended) The computer program product of claim 10, wherein the computer program further includes instructions for causing a computer to:

identify ~~identifying~~ one or more subsections, each subsection including one or more contiguous centerscan boundary pixels having the same color, to be used in trapping.

19. (Currently Amended) A computer-implemented method for identifying, in a device space, an effective centerscan object color along an edge between an overscan object and a centerscan object, the centerscan object having a higher paint order than the overscan object, the method comprising:

mapping the edge to the device space;

identifying a set of device space pixels that are intersected by the edge;

determining for each pixel in the set of pixels if a center of the pixel maps to the centerscan object;

identifying the pixel as a centerscan boundary pixel if the center of the pixel maps to the centerscan object;

identifying the pixel as an overscan boundary pixel if the center of the pixel does not map to the centerscan object;

computing ~~creating a vector pointing in~~ a direction of the centerscan object relative to the edge, the direction indicating on which side of the edge the centerscan object lies;

applying the ~~vector~~ direction to each identified overscan boundary pixel to identify a corresponding centerscan boundary pixel, each centerscan boundary pixel being a device space pixel on the centerscan object side of the edge that is adjacent to an overscan boundary pixel ~~to each identified overscan boundary pixel~~; and

mapping each centerscan boundary pixel to the centerscan object to identify a color of the centerscan boundary pixel.

20. (Original) The method of claim 19, wherein the centerscan object is a raster image and the overscan object is a vector object.

21. (Currently Amended) The method of claim 20, wherein ~~an image resolution differs from a device resolution~~ a size of pixels in the centerscan object is different than a size of the centerscan boundary pixels when each centerscan boundary pixel is mapped to the centerscan object.

22. (Currently Amended) The method of claim 19, wherein ~~identifying a~~ mapping each centerscan boundary pixel to the centerscan object to identify the color of the centerscan boundary pixel comprises:

assigning a color to ~~coloring~~ the centerscan boundary pixel ~~in the device space in accordance with a center scan rule~~ using a color of the centerscan object at the center of the centerscan boundary pixel.

23. (Currently Amended) The method of claim 19, wherein ~~creating a vector~~ computing a direction comprises:

creating a vector specified in device pixels.

24. (Currently Amended) The method of claim 19, wherein ~~creating a vector~~ computing a direction comprises:

creating a vector normal to the edge.

25. (Currently Amended) The method of claim 19, wherein ~~creating a vector~~ computing a direction comprises:

creating a vector normal to an axis in the device space.

26. (Cancelled)

27. (Currently Amended) The method of claim 19, further comprising:

identifying one or more subsections, each subsection including one or more contiguous centerscan boundary pixels having the same color, to be used in trapping.

28. (Currently Amended) A computer program product, residing on a computer-readable medium, for identifying, in a device space, an effective centerscan object color along an edge between an overscan object and a centerscan object, the centerscan object having a higher paint order than the overscan object, the computer program product ~~containing~~ comprising instructions for causing a computer to:

map the edge to the device space;

identify a set of device space pixels that are intersected by the edge;

determine for each pixel in the set of pixels if a center of the pixel maps to the centerscan object;

identify the pixel as a centerscan boundary pixel if the center of the pixel maps to the centerscan object;

identify the pixel as an overscan boundary pixel if the center of the pixel does not map to the centerscan object;

compute ~~create a vector pointing in~~ a direction of the centerscan object relative to the edge, the direction indicating on which side of the edge the centerscan object lies;

apply the ~~vector~~ direction to each identified overscan boundary pixel to identify a corresponding centerscan boundary pixel, each centerscan boundary pixel being a device space pixel on the centerscan object side of the edge that is adjacent to an overscan boundary pixel ~~to each identified overscan boundary pixel~~; and

map each centerscan boundary pixel to the centerscan object to identify a color of the centerscan boundary pixel.

29. (Original) The computer program product of claim 28, wherein the centerscan object is a raster image and the overscan object is a vector object.

30. (Currently Amended) The computer program product of claim 29, wherein ~~an image resolution differs from a device resolution~~ a size of pixels in the centerscan object is different than a size of the centerscan boundary pixels when each centerscan boundary pixel is mapped to the centerscan object.

31. (Currently Amended) The computer program product of claim 28, wherein the computer program further includes instructions for causing a computer to:

assign a color to each [[the]] centerscan boundary pixel ~~in the device space in accordance with a center scan rule~~ using a color of the centerscan object at the center of each respective centerscan boundary pixel.

32. (Currently Amended) The computer program product of claim 28, wherein the ~~computer program further includes~~ instructions for causing a computer to compute a direction include instructions for causing a computer to:

create a vector specified in device pixels.

33. (Currently Amended) The computer program product of claim 28, wherein the ~~computer program further includes~~ instructions for causing a computer to compute a direction include instructions for causing a computer to:

create a vector normal to the edge.

34. (Currently Amended) The computer program product of claim 28, wherein the ~~computer program further includes~~ instructions for causing a computer to compute a direction include instructions for causing a computer to:

create a vector normal to an axis in the device space.

35. (Cancelled)

36. (Currently Amended) The computer program product of claim 28, wherein the computer program further includes instructions for causing a computer to:

identify ~~identifying~~ one or more subsections, each subsection including one or more contiguous centerscan boundary pixels having the same color, to be used in trapping.

37. (Currently Amended) A computer-implemented method for identifying, in a device space, an effective color along one side of an edge between a first centerscan object and a second centerscan object, the method comprising:

mapping the edge to the device space;

identifying a set of device space pixels that are intersected by the edge;

identifying a pixel in the set of device space pixels as a first object boundary pixel if the center of the pixel maps to the first centerscan object;

identifying a pixel in the set of device space pixels as a second object boundary pixel if the center of the pixel does not map to the first centerscan object;

computing ~~creating a vector pointing in~~ a direction of the second centerscan object relative to the edge, the direction indicating on which side of the edge the second centerscan object lies;

applying the ~~vector~~ direction to each identified first object boundary pixel to identify a corresponding second object boundary pixel, each second object boundary pixel being a device space pixel on the second centerscan object side of the edge that is adjacent to a first object boundary pixel ~~to each identified first object boundary pixel~~; and

mapping each second object boundary pixel to the second centerscan object to identify a color of the second object boundary pixel.

38. (Original) The method of claim 37, wherein at least one of the first centerscan object and the second centerscan object is a raster image.

39. (Currently Amended) The method of claim 38, wherein ~~an image resolution differs from a device resolution~~ a size of pixels in the second centerscan object is different than a size of the second object boundary pixels when each second object boundary pixel is mapped to the second centerscan object.

40. (Currently Amended) The method of claim 37, wherein ~~identifying a~~ mapping each second object boundary pixel to the second centerscan object to identify the color of the second object boundary pixel comprises:

assigning a color to the second object boundary pixel ~~in the device space in accordance with a center scan rule~~ using a color of the second centerscan object at the center of the second object boundary pixel.

41. (Currently Amended) The method of claim 37, wherein ~~creating a vector~~ computing a direction comprises:

creating a vector specified in device pixels.

42. (Currently Amended) The method of claim 37, wherein ~~creating a vector~~ computing a direction comprises:

creating a vector normal to the edge.

43. (Currently Amended) The method of claim 37, wherein ~~creating a vector~~ computing a direction comprises:

creating a vector normal to an axis in the device space.

44. (Cancelled)

45. (Original) The method of claim 37, further comprising:

identifying one or more subsections, each subsection including one or more contiguous centerscan boundary pixels having the same color, to be used in trapping.

46. (Currently Amended) A computer program product, residing on a computer-readable medium, for identifying, in a device space, an effective color along one side of an edge between a first centerscan object and a second centerscan object, the computer program product ~~containing~~ comprising instructions for causing a computer to:

map the edge to the device space;

identify a set of device space pixels that are intersected by the edge;

identify a pixel in the set of device space pixels as a first object boundary pixel if the center of the pixel maps to the first centerscan object;

identify a pixel in the set of device space pixels as a second object boundary pixel if the center of the pixel does not map to the first centerscan object;

compute ~~create a vector pointing in~~ a direction of the second centerscan object relative to the edge, the direction indicating on which side of the edge the second centerscan object lies;

apply the ~~vector~~ direction to each identified first object boundary pixel to identify a corresponding second object boundary pixel, each second object boundary pixel being a device space pixel on the second centerscan object side of the edge that is adjacent to a first object boundary pixel ~~to each identified first object boundary pixel~~; and

map each second object boundary pixel to the second centerscan object to identify a color of the second object boundary pixel.

47. (Original) The computer program product of claim 46, wherein at least one of the first centerscan object and the second centerscan object is a raster image.

48. (Currently Amended) The computer program product of claim 47, wherein ~~an image resolution differs from a device resolution~~ a size of pixels in the second centerscan object is different than a size of the second object boundary pixels when each second object boundary pixel is mapped to the second centerscan object.

49. (Currently Amended) The computer program product of claim 46, wherein the computer program further includes instructions for causing a computer to:

assign a color to [[the]] each second object boundary pixel using a color of the second centerscan object at the center of each respective second object boundary pixel ~~in the device space in accordance with a centerscan rule~~.

50. (Currently Amended) The computer program product of claim 46, wherein the ~~computer program further includes~~ instructions for causing a computer to compute a direction include instructions for causing a computer to:

create a vector specified in device pixels.

51. (Currently Amended) The computer program product of claim 46, wherein the ~~computer program further includes~~ instructions for causing a computer to compute a direction include instructions for causing a computer to:

create a vector normal to the edge.

52. (Currently Amended) The computer program product of claim 46, wherein the ~~computer program further includes~~ instructions for causing a computer to compute a direction include instructions for causing a computer to:

create a vector normal to an axis in the device space.

53. (Cancelled)

54. (Currently Amended) The computer program product of claim 46, wherein the computer program further includes instructions for causing a computer to:

identify ~~identifying~~ one or more subsections, each subsection including one or more contiguous centerscan boundary pixels having the same color, to be used in trapping.